**Validation**

### **1. Account and Profile Management**

* **Registration**:
* ***Email*:** Must be in valid format and unique**.**
* ***Password***: At least 8 characters, mixed types for security.
* ***Phone*** *Number*: Validate numeric format, with country code.
* **Social Media Registration**: Validate that data from social media accounts is correct (achievable with basic API setup).
* **Login**: Ensure valid formats for email and password.
* **Verification**: Send a code to email/phone for security.
* **Updating Personal Info**: Ensure valid formatting for names, addresses, and numbers.
* **Address** must contain sufficient information for shipping (e.g., street, city, postal code).
* **Implementation possibility**: The development team can do that.

As for the user, he must follow these instructions.

### **2. Product Interaction**

* **Product Display**:
  + Ensure product details like name, price, quantity are present(Simple).
  + *Price & Quantity*: Prices >0; quantity must be positive integer(Basic checks, feasible).
* **Filtering**: Ensure filters like price, brand, and rating are valid and match available values.
* **Reviews and Ratings**:
  + *Ratings*: Numeric, between 1-5.
  + *Comments*: Limit length, no bad words.

**Implementation possibility**: Easy to implement.

### **3. Shopping Cart and Checkout**

* **Cart Management**:
  + Check stock availability for requested quantity.

**Implementation possibility**: Easy with backend team.

* **Checkout Process**:
  + *Address*: Ensure complete for delivery.
  + *Payment*: Validate card details and formats.

**Implementation possibility**: Possible with form validation and basic payment API

### 4. Order Management

* **Order Tracking**: Display correct order stage (e.g., Processing, Shipped).
* **Implementation possibility**: Easy, feasible with backend team.
* **Return Requests**: Validate eligibility based on policy.
  + **Implementation possibility**: we can do this with Customer Support team.
* **Past Orders**: Ensure order details are correct.
  + **Implementation possibility**: Feasible with correct data storage

### **5. Customer Support**

**Live Chat and Support :**

* + Ensure text inputs are free from inappropriate language and function correctly.
  + **Implementation possibility**: It is possible by providing an interactive chat or social media.

### **6. Engagement and Promotions**

* **Discount Codes**:
  + *Code Validity*: Check expiration and conditions.
  + *Applicability*: Verify code meets item conditions.

**Implementation possibility**: Achievable with basic backend rules and logic.

### **7. Performance and Scalability**

* **High Concurrent User Support**: Ensures site stability under heavy user load.
* **Implementation possibility**: Difficult for us; requires advanced infrastructure.
* **Updates**: Automatically updates stock after purchases.
* **Implementation possibility**: Possible with a simple database setup.

### **8. Security**

* **Data Encryption**: Secures passwords and payment details.
* **Implementation possibility**: Achievable with basic encryption methods.
* **Role-Based Access Control**: Restricts access based on roles (e.g., user, admin).

**Implementation possibility**: Possible with simple backend configurations.

### **9. Reliability and Uptime**

* **Continuous Availability**: Ensures site remains accessible to users at all times.
* **Implementation possibility**: requires strong hosting(we can't do this).
* **Load Balancing**: Distributes user traffic to prevent overload.

**Implementation possibility**: needs advanced server management(we can't do this).

### **10. Compliance and Policies**

* **Privacy Compliance**: Collects user consent for data processing.
* **Implementation possibility**: Possible by adding a consent option during registration.
* **Product Compliance Verification**: Rejects unapproved or restricted products.

**Implementation possibility**: Achievable with basic rules in product management.

### **11. No Guest Checkout**

* Ensure that the checkout process requires user login, and guest checkout is not allowed (customer need this ).
* **Implementation Possibility: we** can implement this with basic **PHP or JavaScript** session management.

### **12. No Unsecured Payment Processing**

* Ensure that payments are only processed through secure, PCI DSS-compliant payment gateways, and data is encrypted during the transaction process (customer need this ) .
* **Implementation Possibility**:We can use API.

### **13. No Unauthorized Access to Admin Features**

* Ensure that admin features such as order management and inventory are only accessible by authorized users with "admin" roles.
* **Implementation Possibility**: We can start by defining roles (like " user" and "admin")

### **14. No User-to-User Communication**

* Ensure there are no direct communication channels (e.g., direct messaging or chat) between users.
* **Implementation possibility**: We will not design chats between users

### **15. No Modification of Critical Data by Users**

* Ensure users cannot modify order details or transaction records after the transaction is complete.
* **Implementation possibility**: We can do this by adding a field in our database

For example, a field can be added to the database that specifies whether or not the data can be modified after the order is completed.